Organisation efficace des données en mémoire pour un décodage rapide d'images fixes

A. Zergaïnoh[*+], P. Duhamel[*]

[*]L2TI, Institut Galilée, Université Paris 13,
99, Avenue Jean Baptiste Clément, 93 430 Villetaneuse, France

[+]LSS/CNRS, Supélec,
Plateau de Moulon, 91 192 Gif sur Yvette, France

email: {zergainoh, duhamel}@lss.supelec.fr

Résumé

Cet article présente un nouveau schéma de codage pour des images fixes. Le codeur est basé sur le principe de sous-échantillonnage adaptatif de l'image à coder. Le décodeur reconstruit les pixels retirés par le codeur à partir d'un algorithme de prédiction. Nous proposons plusieurs modifications de cet algorithme de façon à réduire le coût de codage et le temps de calcul de décodage. Pour réduire le coût de codage, nous construisons un modèle théorique simple des covariances approximant au mieux les covariances de l'image à coder. Quant au temps de calcul de décodage, nous proposons une organisation efficace en mémoire des covariances ainsi que des distances pour la recherche rapide des plus proches voisins. Les simulations effectuées sur des images de test, montrent que le codec présente des résultats compétitifs aux meilleurs codeurs actuels disponibles dans la littérature à savoir JPEG 2000 et SPIHT avec codage arithmétique.

Mots clefs

Echantillonnage adaptatif irrégulier, prédiction, covariance, codage entropique.

1. Introduction

L'état de l'art en codage d'images fixes montrent que les meilleurs codeurs actuellement disponibles sont essentiellement basés sur les transformées en ondelettes. Pour ce type de codeur, trois principaux modules sont mis en œuvre. Le premier concerne essentiellement la transformée en ondelette qui décorrèle les coefficients de l'image, le second s'intéresse à la quantification de ces coefficients et le troisième concerne le codage entropique. Parmi les meilleurs codeurs d'images actuels, nous pouvons citer les codeurs basés sur les algorithmes EZW ([1]), SPIHT ([2]) et EBCOT ([3], [4]) adopté par la norme de compression des images fixes JPEG2000 ([5]). De nombreux travaux de recherche développés sur cette thématique se sont plutôt penchés sur l'étude et l'amélioration de chaque module.

Cet article propose une nouvelle approche de codage d'image basée sur l'échantillonnage irrégulier de l'image originale. Généralement une image numérique contient des zones homogènes représentant des régions lisses et des zones inhomogènes correspondant à des régions de détail. Pour échantillonner une image, habituellement une période d'échantillonnage au dessus de celle de Nyquist est choisie. Celle-ci permet d'extraire de manière uniforme les pixels appartenant à l'image. Cependant la période d'échantillonnage appliquée aux zones lisses peut s'avérer trop petite par rapport à la période choisie d'autant plus si le contenu de l'image est très hétérogène. Nous proposons dans cet article d'adapter la période d'échantillonnage aux différentes zones de l'image afin de réduire dans un premier temps la redondance de l'information de l'image à coder. Les pixels retenus sur l'image doivent suffire pour permettre une bonne reconstruction de l'image originale au niveau du décodeur. Un algorithme de sous-échantillonnage adaptatif basé sur le critère de la minimisation de l'erreur de reconstruction est proposé.

Cette reconstruction est basée sur un algorithme de prédiction. Il existe dans la littérature de nombreux algorithmes de reconstruction ([7], [8], [9], [10]). Néanmoins certains algorithmes sont coûteux en temps de calcul et nécessitent un coût de codage élevé pour une qualité de reconstruction moyenne.

Les performances de notre codec sont étroitement liées aux différentes modifications apportées à l'algorithme de prédiction présenté dans ([12]). En effet, pour réduire le coût de codage, nous proposons de modéliser les covariances de l'image par un modèle théorique très simple évitant ainsi de coder les covariances puis de les transmettre au décodeur. Pour réduire le temps de calcul prohibitif lié au décodage, nous proposons une solution qui consiste à organiser de manière astucieuse les données de covariance en mémoire. De part cette stratégie, les valeurs sont directement disponibles en mémoire durant tout le traitement. Quant à la recherche des plus proches voisins, nous proposons d'organiser les distances

euclidiennes en mémoire de façon à pouvoir y accéder à tout moment pendant le traitement de décodage.

L'approche globale proposée est récapitulée par la Figure 1. L'article est organisé comme suit. La section 2 présente l'algorithme de sous-échantillonnage. La section 3 introduit l'algorithme de prédiction et ses inconvénients. La section 4 propose les diffréntes modifications apportées à l'algorithme de prédiction. La section 5 fournit les résultats de simulation. La section 6 conclut notre travail.

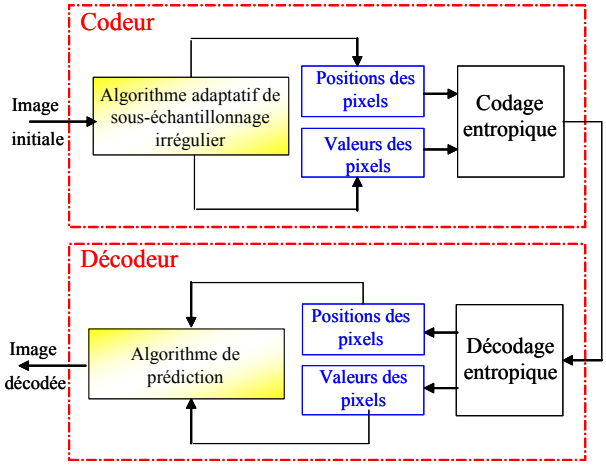

Figure 1 - Schéma du codec proposé pour des images fixes

2. Algorithme adaptatif de sous-échantillonnage irrégulier

Le codeur agit directement sur les pixels de l'image. Il retire de l'image à coder les pixels jugés redondants. Le processus d'extraction est réalisé par un algorithme de sous-échantillonnage adaptatif. Celui-ci optimise la sélection des pixels de façon à réduire l'erreur quadratique moyenne de reconstruction. L'algorithme d'optimisation de sous-échantillonnage proposé est étroitement lié à l'algorithme de prédiction utilisé par le décodeur.

L'algorithme procède en deux étapes. Supposons que l'on veuille garder uniquement η pixels sur l'image à coder.

L'algorithme retient tout d'abord un nombre très faible de pixel, de l'ordre de $\eta_0 << \eta$, sur l'image. Ces pixels sont uniformément distribués sur l'image. La reconstruction de chaque pixel retiré de la grille de l'image est obtenue à partir d'un algorithme de prédiction qui sera introduit à la section suivante. Afin d'atteindre les η pixels à retenir sur l'image, l'algorithme de sous-échantillonnage adaptatif sélectionne les pixels qui engendrent en chaque pixel prédit une erreur de reconstruction maximale.

La deuxième étape consiste à affiner le choix des pixels retenus sur l'image. En effet, les premiers pixels η_0 répartis uniformément sur la grille ne sont pas forcément ceux qui introduisent la plus grande erreur de reconstruction. Chaque pixel appartenant à la grille irrégulière est remplacé par un de ses huit voisins. Pour chacune des huit permutations, l'image est reconstruite localement. L'algorithme de sous-échantillonnage retient le pixel pour lequel l'erreur quadratique moyenne calculée localement est minimale. Cette opération est itérée plusieurs fois sur l'image entière jusqu'à ce que l'algorithme converge vers une erreur quadratique moyenne stable.

3. Présentation de l'algorithme de prédiction et ses inconvénients

Avant de présenter l'algorithme de prédiction et ses inconvénients, introduisons quelques notations.

Notons I l'image originale à coder. Elle est définie comme suit

$$I = \{u(x_i, y_i) \text{ avec } x_i = 1,...,M; y_i = 1,...,N\} \quad (1)$$

où $u(x_i, y_i)$ représente le niveau de gris du pixel localisé en (x_i, y_i) sur la grille de l'image I.

Notons I_0 l'image irrégulièrement échantillonnée. La prédiction du niveau de gris d'un pixel de coordonnées (x_i, y_i) sur l'image originale I est note $\hat{u}(x_i, y_i)$.

3.1. Algorithme de prédiction

L'estimation du niveau de gris d'un pixel inconnu, localisé en (x_k, y_k) sur l'image irrégulièrement échantillonnée est donnée par la combinaison linéaire pondérée des n plus proches pixels dans un voisinage noté D_{x_k,y_k} autour du pixel à estimer. L'estimateur est donné par la relation suivante :

$$\hat{u}(x_k, y_k) = \sum_{i=1}^{n} a_{i,D_{x_k,y_k}} u(x_i, y_i)$$

$$\text{avec } u(x_i, y_i) \in D_{x_k,y_k} \quad (2)$$

où $a_{i,D_{x_k,y_k}}$ est le i-ème coefficient associé au pixel $u(x_i, y_i)$ de coordonnées (x_i, y_i) sur la grille de l'image I_0.

Le meilleur estimateur linéaire non-biaisé est obtenu en déterminant les coefficients $a_{i,D_{x_k,y_k}}$ qui minimisent l'erreur quadratique moyenne entre l'image originale et celle prédite. Le formalisme de Lagrange nous amène à résoudre le système d'équation suivant :

$$\begin{cases} \sum_{i=1}^{n} a_{i,D_{x_k,y_k}} Cov(u(x_j, y_j), u(x_i, y_i)) + \mu_{D_{x_k,y_k}} = \\ \qquad = Cov(u(x_j, y_j), u(x_k, y_k)) \quad \forall j \in N \\ \sum_{i=1}^{n} a_{i,D_{x_k,y_k}} = 1 \end{cases} \quad (3)$$

où $Cov(u(x_j, y_j), u(x_i, y_i))$ représente la covariance entre deux pixels $u(x_i, y_i)$ et $u(x_j, y_j)$ localisés respectivement en (x_i, y_i) et $u(x_j, y_j)$ sur la grille de l'image originale. $\mu_{D_{x_k,y_k}}$ est le multiplieur de Lagrange.

3.2. Inconvénients majeurs

Evaluons les ressources requises par cet algorithme. Commençons par le point de vue coût de codage. A chaque fois que le décodeur doit prédire le niveau de gris d'un pixel, il doit disposer des informations suivantes :

• les coordonnées des pixels retenus sur la grille de l'image ;
• les niveaux de gris des pixels retenus sur la grille de l'image ;
• les valeurs des covariances de l'image.

Quant au temps de décodage, à chaque fois que le décodeur prédit le niveau de gris d'un pixel, le calcul des $a_{i,D_{x_k,y_k}}$ s'impose. Dans ce but le décodeur procède comme suit :
• recherche des n plus proches voisins autour du pixel à prédire;
• construction de la matrice des covariances du système d'équation (3);
• résolution du système d'équation (3) qui consiste à inverser la matrice de covariance de taille $(n+1)\times(n+1)$.

L'utilisation de l'algorithme de prédiction ainsi présenté affecte considérablement le coût de codage de la méthode de compression. De plus, le temps de calcul global de décodage est prohibitif. Des modifications de l'algorithme de prédiction s'avèrent donc indispensables afin d'atteindre des performances comparables aux codeurs actuels.

4. Algorithme de prédiction modifié pour un décodage rapide

Pour détourner les inconvénients de l'algorithme de prédiction précédemment présenté, nous proposons plusieurs modifications de l'algorithme que nous listons ci-dessous.

4.1.Réduction du nombre de voisins

Le nombre de voisins requis par l'algorithme de prédiction est restreint à 3 pixels. Cette réduction du nombre de voisins n'affecte pas la qualité de l'image reconstruite (méthode de reconstruction locale). Ce choix se répercute directement sur la taille des matrices à manipuler. L'équation (2) s'écrit alors comme suit :

$$\hat{u}(x_k,y_k)=\sum_{i=1}^{3} a_{i,D_{x_k,y_k}}\,u(x_i,y_i) \quad \text{avec } u(x_i,y_i)\in D_{x_k,y_k}$$

Le calcul des coefficients de pondération $a_{i,D_{x_k,y_k}}$ est ramener sous une forme matricielle :

$$\begin{bmatrix} a_{1,D_{x_k,y_k}} \\ a_{2,D_{x_k,y_k}} \\ a_{3,D_{x_k,y_k}} \\ \mu_{D_{x_k,y_k}} \end{bmatrix} = \begin{bmatrix} 1 & Cov(h_{1,2}) & Cov(h_{1,3}) & 1 \\ Cov(h_{2,1}) & 1 & Cov(h_{2,3}) & 1 \\ Cov(h_{3,1}) & Cov(h_{3,2}) & 1 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} Cov(h_{1,k}) \\ Cov(h_{2,k}) \\ Cov(h_{3,k}) \\ 1 \end{bmatrix} \quad (4)$$

où $Cov(h_{i,j})=Cov(h_{j,i})=Cov(u(x_j,y_j),u(x_i,y_i))$.

Des détails complémentaires sur ces dernières notations sont donnés ci-dessous.

4.2.Construction d'un modèle théorique de covariance

Inspiré des méthodes de Krigeage développées dans des applications de géologie ([6]), nous proposons de construire un modèle théorique de covariance qui approxime au mieux les covariances expérimentales de l'image. Dans ce cas le codeur s'affranchit de l'envoi des valeurs de covariances expérimentales au décodeur. Le coût de codage est ainsi réduit puisqu'il ne concerne que les pixels retenus sur la grille de l'image ainsi que leurs positions respectives.

L'étape de modélisation est très importante puisque le modèle choisi influe sur la qualité de l'image reconstruite. De nombreuses simulations ont été réalisées pour analyser et rechercher les modèles théoriques adéquats de covariance à adapter aux covariances expérimentales de l'image. Dans la plupart des cas, le modèle linéaire est très approprié si on se restreint à un domaine où la portée est inférieure à environ 20 pixels.

Nous proposons d'utiliser le modèle suivant où h représente dans ce cas la distance Euclidienne normalisée par rapport à la portée choisie :

$$Cov(h)=1-h \quad (5)$$

4.3. Algorithme de recherche rapide des plus proches voisins

La recherche des plus proches voisins est très coûteuse en terme de temps de calcul. En effet, pour chaque pixel à prédire, le décodeur doit chercher tous les pixels proches du pixel à prédire. Ce sont les distances euclidiennes entre pixels qui sont utilisées pour mesurer l'éloignement des pixels.

L'algorithme proposé travaille sur une fenêtre glissante dont la taille est fixée par la portée choisie :

$$D_{porte,porte}=[0,2\times porte]\times[0,2\times porte]$$

Comme phase d'initialisation, l'algorithme classe les $(2\times porte+1)^2$ distances euclidiennes notées $h_{i,j}$ dans un ordre croissant. Ces distances sont calculées par rapport au centre de la fenêtre $(porte,porte)$ c'est à dire la position du pixel à prédire. Les distances $h_{i,j}$ sont stockées dans la première colonne. Ce classement n'est effectué qu'une seule fois et sera utilisé pendant tout le traitement de décodage. La deuxième et troisième colonnes sont ensuite mises à jour.

Pour une recherche rapide, nous proposons l'organisation mémoire illustrée par la Figure 2.

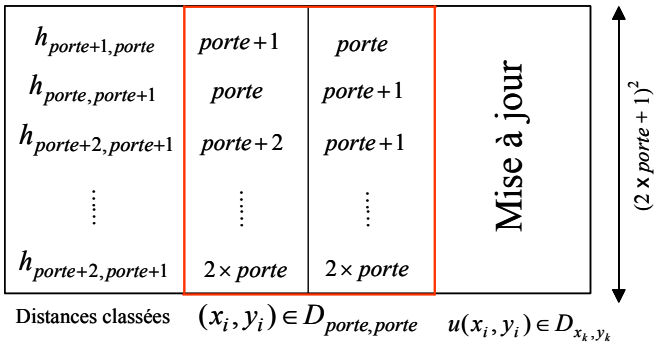

Figure 2 - Algorithme de recherche rapide des plus proches voisins

Pour la prédiction d'un pixel quelconque $\hat{u}(x_l, y_l)$, l'algorithme met à jour la deuxième colonne comme suit:

$$\begin{cases} x_i = x_i + x_l - porte \\ y_i = y_i + y_l - porte \end{cases} \text{ avec } (x_l, y_l) \in D_{x_l, y_l} \qquad (6)$$

La troisième colonne concerne les valeurs des pixels retenues sur la grille de l'image sous-échantillonnée. Cette colonne est mise à jour en accordance avec la deuxième colonne. Les pixels qui n'ont pas été retenus sur la grille de l'image sont représentés par une chaîne de caractère par exemple « Pabs » pour indiquer « pixel absent ». Puisque les distances sont déjà classées, le décodeur n'aura qu'à récupérer les 3 plus proches pixels disponibles dans le tableau.

4.4. Organisation mémoire des covariances

Basé sur le modèle linéaire de covariance, le décodeur propose de calculer et d'organiser les covariances en mémoire une seule fois de façon à pouvoir y accéder à chaque prédiction de pixel, sans avoir à les calculer. Cette stratégie permet de réduire significativement le temps de calcul.

Pour une portée fixée, les covariances théoriques sont calculées sur la fenêtre glissante précédemment définie selon la relation suivante :

$$Cov(h_{i,j}) = 1 - h_{i,j} \text{ pour } i, j \in [0, 2 \times porte] \qquad (7)$$

Ces covariances sont judicieusement stockées en mémoire. L'organisation est illustrée par la Figure 3.

Les expressions mathématiques des coefficients de pondération de l'équation (4) sont déterminées une fois pour toute, évitant ainsi l'inversion matricielle à chaque prédiction.

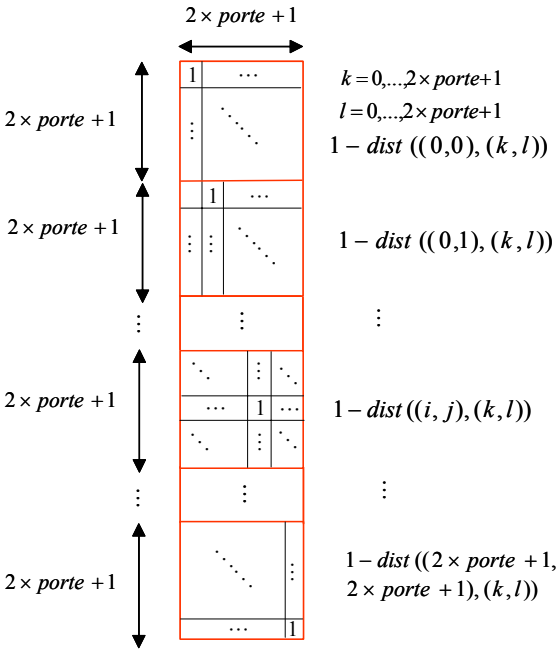

Figure 3 - Organisation efficace des covariances en mémoire

5. Résultats expérimentaux

En ce qui concerne le codage des coordonnées des pixels retenus, une carte des coordonnées est construite. Celle-ci est de même taille que l'image originale. Sur cette carte, les pixels retenus sont représentés par « 1 » en leurs positions respectives et un « 0 » pour indiquer l'absence de pixels. Cette carte à deux niveaux ainsi que les valeurs des pixels retenus sont codés par le biais d'algorithmes de codage entropique.

Nous codons la carte des positions par le codage entropique JBIG2 dont le programme est fourni à l'adresse Internet donnée par [13]. Quant aux valeurs des pixels, nous proposons de les coder par le biais de l'algorithme LZMA (Lempel-Ziv Chian Markov Algorithm) dont le programme est fourni à l'adresse indiquée par [16].

Le temps de calcul de l'algorithme de prédiction sans organisation mémoire est prohibitif de l'ordre de l'heure sur un ordinateur à 2.5 Ghz. L'organisation des données en mémoire nous a permis de ramener ce temps de calcul à quelques secondes.

La qualité des images décodées est mesurée par le rapport signal à bruit (PSNR) calculé entre l'image originale et l'image décodée. Les performances de notre codec sont comparées par rapport aux meilleurs codeurs SPIHT et JPEG2000. Les programmes utilisés sont disponibles aux pointeurs Internet respectifs [14] et [15]. Les simulations sont réalisées sur les différentes images de test Lena, boat, Goldhill et peppers.

L'algorithme de sous-échantillonnage adaptatif est appliqué sur chaque image de test. L'échantillonnage η, sur chaque image de test, est fixé à $\eta = 4.9\%$ avec un échantillonnage initial η_0 égal à 0.49%.

Le Tableau 1 compare les performances des différents codeurs. Pour le codeur proposé, les résultats sont similaires pour Lena, moins bons pour l'image boat et meilleurs pour les images peppers et Goldhill. En effet, les résultats sont étroitement liés à la complexité de l'image. Le codec proposé est bien adapté aux images dans lesquelles apparaissent de nombreuses zones plus ou moins lisses (exemples peppers et Goldhill). La Figure 4 présente l'image peppers sous-échantillonnée ainsi que sa reconstruction.

Pour des débits moyens, les résultats présentés montrent que le codec proposé est compétitif aux meilleurs codeurs. Par contre pour des débits binaires très bas, c'est à dire un échantillonnage η très faible ($\eta < 1\%$), la qualité de l'image décodée ne sera pas acceptable pour ce type d'images de test.

6. Conclusion

Nous avons proposé dans cet article, un nouveau codec pour des images fixes. Le codeur est basé sur un algorithme de sous-échantillonnage adaptatif de l'image basé sur le critère de la minimisation de l'erreur de

reconstruction. Les performances de notre codec, en terme de coût de codage et de temps de calcul de décodage, sont étroitement liées aux différentes modifications apportées à l'algorithme de prédiction. En effet, pour réduire le coût de codage, nous avons proposons de modéliser les covariances de l'image par un modèle théorique très simple évitant ainsi de coder et de transmettre les covariances au décodeur. En ce qui concerne la réduction significative du temps de calcul de décodage, nous avons proposé une solution qui consiste à organiser de manière astucieuse les données de covariance en mémoire. De part cette stratégie, les valeurs sont directement disponibles en mémoire durant tout le traitement. Quant à la recherche des plus proches voisins, nous avons organisé les distances en mémoire de façon à pouvoir y accéder directement. Les simulations réalisées sur des images de tests montrent que notre codec présentent des résultats compétitifs aux meilleurs codeurs actuels pour des débits binaires moyens.

References

[1] J. M. Shapiro, Embedded image coding using zerotrees of wavelet coefficients, *IEEE Transactions on signal processing,* 41(12) : 3445-3462, 1993.

[2] A. Said, W. A. Pearlman, A new fast and efficient image codec based on set partitionning in hierarchical trees, *IEEE Transactions on circuits and systems for video technology*, 6(3) : 243-250, 1996.

[3] D. Taubman, High performance scalable image compression with EBCOT, *IEEE Transactions on Image Processing*, 9(7) : 1151-1170, July 2000.

[4] D. Taubman, E. Ordentlich, M. Weinberger, G. Seroussi, Embedded block coding in JPEG 2000, *Signal Processing-Image Communication*, 17(1) : 49-72, January 2002.

[5] ISO/IEC 15444-1: JPEG2000 image coding system, 2000.

[6] D. G. Krige, A statistical approach to some basic mine valuation problems on the Witwatersrand, *Journal chem. Metal. Min. S. Afr*., S2, pp. 119-139, 1951.

[7] L. L. Schumaker, Fitting Surfaces to Scattered Data, pp. 203-268, G., Lorentz, Georg Gunter, eds. Chui, Schumaker, 1976.

[8] R. Franke, G. Nielson, Smooth interpolation of large sets of scattered data, *International of Numerical Methods in Engineering*, vol. 15, 1691-1704.January 1980.

[9] D. Shepard, A two dimensional interpolation function for irregularly spaced data, proceedings 23rd Nat. conference ACM, 517-523, 1968.

[10] H. Le Floch, C. Labit, Irregular image sub-sampling and reconstruction by adaptive sampling, *ICIP*, vol. 3, 379-382, 1996.

[11] M. Babel, O. Déforges, J. Ronsin, L. Bédat, N. Normand, B. Parrein, J.P. Guédon, "Le LAR aux MOJETTES, dans *Actes de la conférence CORESA'04*, Lille, Mai 2004.

[12] A. Zergaïnoh, J.-P. Astruc, Hybrid method for still image coding, *Eusipco*, 5-8 september 2000, Tamper Finland.

[13] http://www.jpeg.org/jbig/index.html

[14] http://www.cipr.rpi.edu/research/SPIHT/spiht3.html

[15] http://www.kakadusoftware.com/

[16] http://www.7-zip.org/sdk.html

Taille	Images	Débit (*bpp*)	SPIHT (*dB*)	JPEG2000 (*dB*)	Notre codec (*dB*)
512×512	Lena	0.636	38.29	38.28	38.21
512×512	Boat	0.654	36.09	35.96	35.30
512×512	Peppers	0.635	36.66	36.63	37.27
512×512	Goldhill	0.646	34.25	34.30	34.61

Tableau 1 Comparaisons des performances des différents codeurs

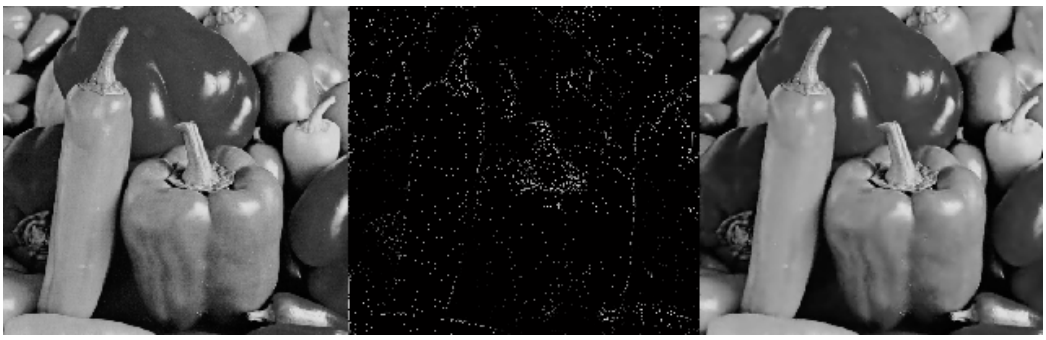

Figure 4 - Image initiale (gauche), image sous-échantillonnée (centre), image reconstruite à partir de 12846 pixels (droite)